

基于语义 Web Services 的模糊匹配

阳 辉,金可音,汤双权,徐利谋

(湖南工业大学 计算机系,湖南 株洲 412008)

摘 要: Web 服务发现过程本质上是用户请求和服务匹配的过程。传统的 UDDI 发现技术是通过精确匹配实现的,不能较好地支持基于概率和语义约束的模糊匹配。针对这种情况,在基于现有的 OWL-S 匹配上,进一步增加了以隶属函数、语义距离、阈值为基础的模糊匹配,形成了两级匹配。并通过对语义表示的服务能力进行模糊聚类,这种机制减少了搜索空间,提高了发现效率及匹配的精度。

关键词: 匹配;隶属函数;语义距离;模糊聚类;OWL-S

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2007)11-0125-03

Fuzzy Matching Based on Semantic Web Services

YANG Hui, JIN Ke-yin, TANG Shuang-quan, XU Li-mou

(Department of Computers, Hunan University of Technology, Zhuzhou 412008, China)

Abstract: The essence of the Web services is the process on matching between the users' requests and the services. As traditional UDDI discovery technology is achieved by precise matching, it can't preferably support the probability and semantic restriction matching. on the basis of the existing OWL-S matching, represents fuzzy matching based on the subjection functions, semantic distances and valve, forming bipod matching. Furthermore, the semantic capacity of services is fuzzily clustered, which decreases searching space and increases the precision.

Key words: matching; subjection functions; semantic distance; fuzzy clustering; OWL-S

0 引言

随着 WWW 的迅猛发展, Web 上的信息量的日益增加,面对内容庞杂、浩瀚如烟的信息,人们面临了一个严峻的问题:如何高效、迅速地找到自己所需服务?建立有效的 Web 发现方法已是非常紧迫的任务。

目前 UDDI 成为 Web 服务的注册和发现机制^[1]。但是 UDDI 存在如下几个缺陷:

(1)对服务的描述缺乏灵活性,使得在服务匹配时只能采用简单的基于关键字搜索方法,不能满足服务发现要求;

(2)使用 XML 来描述数据模式,因为缺少对服务内容的语义描述,机器不能理解,这种描述很难将子服务区分开来,因此它成为 UDDI 发现机制的一个障碍;

(3)不能度量候选者和查询目标间的符合程度。

此外,在自然科学或社会科学研究中,存在着许多定义不很严格或具有模糊性的概念。所谓模糊性,主要是指客观事物的差异在中间过渡中的不分明性,模糊性概念是没有明确外延的概念。比如概念“胖子”“瘦子”,“高”“矮”等,它们是模糊的,很难精确描述。服务的发现过程中存在人机交互,用户是主动的,可以提出模糊的查询要求,则要求计算机进行模糊匹配。例如,服务请求者查找一个预定旅馆服务,要求价格是低廉的。价格低廉是对“预定旅馆服务”服务的约束行为。而以前 Web 服务的匹配,只重视服务功能的语法,忽略对 Web 服务行为约束时的刻画,导致发现准确率不高。对此,在已有的 OWL-S 的匹配上^[2,3],进一步增加以隶属函数、语义距离、阈值为基础的模糊匹配,形成了两级匹配。这种机制通过对语义表示的服务能力进行模糊聚类,减少了搜索空间,提高了发现效率。模糊逻辑的引入,用量化的观点得到直接的结果。

1 理论基础

1.1 模糊集理论

定义 1^[4] 定义一个模糊集 \tilde{A} , 对任何 x , 其隶属

收稿日期:2007-01-06

基金项目:湖南省自然科学基金资助项目(05JJ30122);湖南高校科研项目(04C720)

作者简介:阳 辉(1984-),女,湖南衡阳人,硕士研究生,研究方向为分布式计算、语义 Web;金可音,教授,硕士生导师,研究方向为分布式计算、工作业务流。

度函数 $\mu_{\tilde{A}}(x) \in [0, 1]$ 是 x 到成员空间的映射, X 是 x 表示的对象集合, 可以用式(1) 来表示模糊集 \tilde{A} 。

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\} \quad (1)$$

定义 2^[4] 最大隶属度 1: 设 $S_1, S_2, \dots, S_n \in F(S)$ 为 n 个服务, $S_0 \in U$ 是待识别的服务, 如果 S_i 满足条件

$$\mu_{S_i}(S_0) = \max\{\mu_{S_1}(S_0), \mu_{S_2}(S_0), \dots, \mu_{S_n}(S_0)\} \quad (2)$$

则认为 S_0 相对隶属于 S_i 。

定义 3^[4] 最大隶属度 2: 设 $A \in F(U)$, $\mu_1, \mu_2, \dots, \mu_n \in U$ 是 n 个满足条件的服务的隶属度, 如果 μ_i 满足条件

$$\mu_{\tilde{A}}(\mu_i) = \max\{\mu_{\tilde{A}}(\mu_1), \mu_{\tilde{A}}(\mu_2), \dots, \mu_{\tilde{A}}(\mu_n)\} \quad (3)$$

则应该优先选用 S_i 。

1.2 弹性匹配

基于 UDDI 的服务发现机制是精确匹配的。而通常用户请求的服务和提供者提供的服务存在一定的出入, 用户请求的输入不一定满足服务的输入, 反之, 服务的输出满足用户的输出。于是文献[5]提出了一种“弹性匹配”。语义匹配要满足匹配引擎应该支持请求者和发布者之间的弹性语义匹配。

弹性匹配按程度分为精确(Exact)、插件(plugin)、包含(subsume)、失败(fail)。插件匹配, 包含匹配用到本体中类和类之间继承关系, 因此弹性匹配可以看成应用本体语义的一种语义匹配。

1.3 语义距离

然而, 以上给出的四类分类机制存在一定的缺陷, 它们不能精确描述服务请求者与提供者匹配的“足够相似”。因此引入语义距离, 与给定的阈值 λ 作为匹配的理论基础, 定量地计算匹配的相似程度。匹配度就是用户请求和服务之间的相似度, 用户请求和服务之间的相似度越高, 此服务的匹配度就越高, 即两服务间语义距离越小, 就说明越匹配。

采用如下语义距离公式(给定服务 A, B):

$$d_m(A, B) = \left\{ \frac{1}{2N} * \sum_{i=1}^n [|\mu_A(x) - \mu_B(x)|^q + |\gamma_A(x) - \gamma_B(x)|^q] \right\}^{1/q}, q \geq 1 \quad (4)$$

将语义距离转换成相应的匹配度:

$$\delta_{\text{match}}(A, B) = 1 - d(A, B) \quad (5)$$

给定阈值 λ , 如果 $\delta_{\text{match}}(A, B)$ 大于 λ , 则可认为它们是匹配的。

根据以上语义距离公式可以得到两种特殊的匹配:

① 当 $d(A, B) = 0$ 时, 则是完全精确匹配;

② 当 $d(A, B) = 1$ 时, 则匹配失败。

2 基于语义 Web Service 的模糊匹配框架

2.1 模糊匹配框架

服务模糊匹配框架如图 1 所示。主要由模糊转换器(Fuzzy Converter)、匹配引擎(Matching Engine)、OWL-S2UDDI 转换器^[6]、OWL-S Files、UDDI Register 等组成。

其中 OWL-S Files 是用来存储 OWL-S 的档。OWL-S 是基于本体 OWL 的描述语言, 采用明确的、计算机可以理解的语言标识来描述 Web 试图利用语义描述和逻辑推理最终实现服务匹配的自动化和智能化。Service Profile 是 OWL-S 重要组成部分, 除基本描述之外, 使用输入(Input)、输出(Output)、前提条件(Precondition)和结果(Result)等元素描述 Web 服务。OWL-S2UDDI 转换器将包含语义信息的 OWL-S Files 映射到 UDDI 注册器中供用户查找。模糊转换器用于将服务请求的模糊条件进行转换。例如, 服务请求者查找一个预定旅馆服务, 要求价格是低廉的, 而模糊规则中定义 ≤ 50 为“低廉”, 则通过模糊转换器将这个服务的约束行为进行转换。同理, 也将服务提供者发布的服务转换为相统一的“模糊”服务。这主要是为了在匹配的时候进行近似推理, 有利于系统查找到所需要的服务和子服务。匹配引擎(Matching Engine)主要由 OWL-S 匹配、模糊匹配组成。

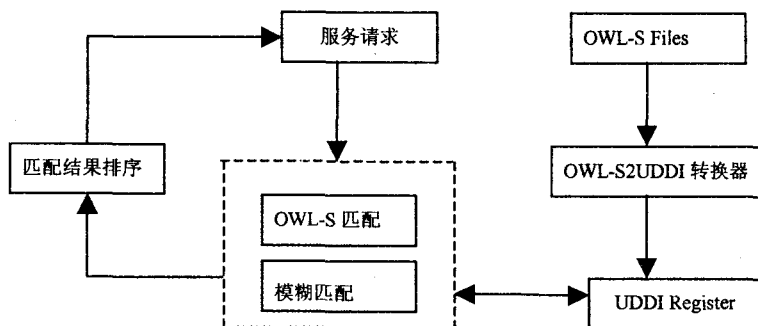


图 1 服务模糊匹配框架

2.2 模糊匹配过程

这里讨论的匹配分为两个层次, 首先是高层的服务能力, 用 Profile 部分来描述服务实体的整体属性, 即基于 OWL-S 的匹配。但是这种匹配对 Web 服务的行为描述信息不足, 只重视服务功能的语法, 忽略对 Web 服务的行为约束的刻画。所以在上述的基础上, 提出了模糊匹配, 只考虑在 Web 服务的约束行为上, 通过增加对约束行为的推理, 来提高匹配的准确率, 即这里讨论的模糊请求。

具体步骤如下:

1) 将要注册的服务转换为能表示服务能力的 OWL-S 语义形式文件, 存储在 OWL-S Files 中。

2) 依据模糊聚类方法按服务能力对服务进行抽象聚类, 形成语义服务的层次结构。基于服务能力的服务聚类方法步骤如下:

(1) 建立模糊相似矩阵 R 。

设 $S = \{S_1, S_2, \dots, S_n\}$ 是 n 个待分类的服务对象, 各自的服务能力有 m 个特征属性, 用 $R = (r_{ij})_{n \times n}$ 来表示模糊相似矩阵, 模糊相似矩阵是一个对角线为 1 的模糊矩阵, r_{ij} 表示服务 S_i 和 S_j 按照 m 个服务能力属性特征相似的程度, 对每一个服务来说都可以用一组数据 $x_{i1}, x_{i2}, \dots, x_{im}$ 来表示它的 m 个特征。采用贴近度距离来求得服务间的相似系数 r_{ij} , 见式(6):

$$r_{ij} = d(\tilde{S}_i, \tilde{S}_j) \quad (6)$$

因此模糊矩阵表示为:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}_{n \times m} \quad (7)$$

(2) 服务聚类。

模糊聚类: 对满足传递性的模糊分类关系的 R^* 进行聚类处理, 给定不同置信水平的 λ , 求 R_λ^* 阵, 找出 R^* 的 λ 显示, 得到普通的分类关系。当 $\lambda = 1$ 时, 每个样品自成一类, 随 λ 值的降低, 由细到粗。

(3) 对于一个新加入的服务属于已有的何种服务类型, 遵从下列最大隶属原则 1。

3) 由于现有的语义不支持 UDDI 的注册, 而通过 UDDI 将服务注册到注册服务器是目前最通用的方法, 因此需要实现 OWL-S 到 UDDI 之间的映射关系, 通过使用 OWL-S 来扩展 UDDI 以及扩展 UDDI API 就可以将语义表示服务高层的 OWL-S Profile 信息嵌入到 UDDI 的 Advertisement 中实现基于语义的服务注册, 文献[1,2]已经给出了实现了这种机制。

4) 服务匹配: 服务消费者查询服务时候, 首先是基于功能的请求, 通过语义表示的对象和请求对象之间的关联, 来触发 OWL-S 匹配机制。把满足条件的结果送到筛选决策器, 进行第二级匹配, 即通过描述 Web 服务约束行为的模糊条件, 触发模糊匹配机制。以上两者的匹配都是通过计算请求服务和输出服务之间的距离(1.3 定义的距离)来实现匹配的。将语义距离转换成相应的匹配度:

$$\delta_{\text{match}}(A, B) = 1 - d(A, B)$$

给定阈值 λ , 如果用户的请求与相应的匹配度大于 λ , 则可认为它们是匹配。否则匹配失败。对满足要求的服务, 根据第 1 节的定义(2) 进行排序, 按照顺序将结果返回给用户。

2.3 匹配算法的描述

根据 2.2 节步骤 3, 做如下规定: req 表示服务请求, Adv 是注册表示能力的服务, CapMatch 表示满足最低隶属度 λ 匹配的服务, FuzzyMatch 用于模糊匹配。

算法如下:

SubWS = Φ // 第二阶段匹配操作的候选子集

sim = μ

CapMatch = empty list; // 服务能力匹配

For $WS_i \in \text{Adv}$

if $\delta_{\text{match}}(\text{req}, WS_i) > \mu_i$

Then SubWS. Add(WS_i)

End if

End for

Return SubWS

Matched = Φ // 匹配成功的 Web 服务集

FuzzyMatch = empty list; // 模糊匹配

For $WS_i \in \text{SubWS}$

if FuzzyMatch(req, WS_i) = True

Then matched. Add(WS_i)

End if

End for

Sort(matched)

Return Sort(matched)

3 小 结

文中基于语义 Web Services 的模糊匹配框架是在 Kuo-Ming Chao^[7]提出的体系结构上进行改进而来的。在 Kuo-Ming Chao 的基础上, 给出了按照服务请求类型进行不同级别的匹配。且利用了服务间的语义距离作为相似度, 在理论上给出了匹配的算法。在现有语义匹配算法的基础上, 提出了一种有助于提高效率的匹配算法——一种两层过滤的匹配思想, 以此来提高服务与请求间的匹配效率。

模糊集理论符合客观世界模糊对象的本质, 成为新的研究热点。通过模糊匹配机制可以服务搜索的效率, 和提高定位合适满足请求者需要的服务的有效性, 自动选择最合适的服务。

下一步的工作是将作为服务相似度的距离进行改进, 例如可以引入加权系数, 使得匹配度更加精确。同时, 也将进一步完善匹配框架, 尤其是它的推理机制, 并将它应用于具体的实例中去。并通过仿真实验与其他几种匹配方法进行了比较, 得出该方法提高了服务发现的查全率和查准率。有理由相信实现这一套机制的服务匹配将进一步推动服务发现的高效化和自动化。

(下转第 138 页)

3)若空字句 NIL 在 file 中,则新规则是已存规则的逻辑结论,结束;

4)按某种策略在 file 表中寻找可归结的字句对,若存在则归结之,并将归结式并入 file 中,转 2);

5)归结失败,不是已存规则的逻辑结论,退出。

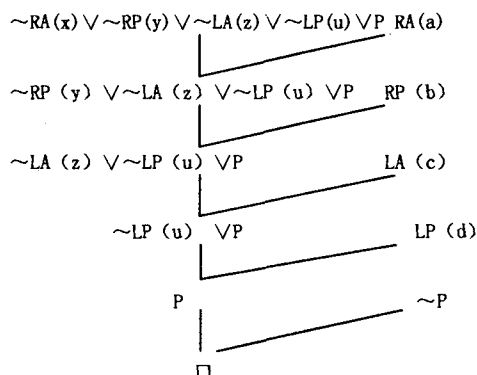


图 2 归结演绎树

4 程序流程

程序首先将刚输入的规则翻译成自然语言,如果觉得不符合要求,继续输入规则。如果用户觉得符合初衷,继续进行判断看该规则是否符合语法检查,如果符合,还要检查该规则是否是前面规则的逻辑结论,如是则相当于存在这条规则。再根据情况做出处理,向用户给出提示。如不是,说明该规则符合要求,把刚刚输入的规则存放到文件中^[5]。流程见图 3。

系统判断某条规则是否存在时,只要判断这条规则是不是前面的规则的逻辑结论即可,不是要判定是否有这样一条完全相同的规则。

规则的自动检测与语义翻译,在实际应用中对网站的管理人员是很有帮助的,可以避免规则的输入错误,或是及早发现错误,以免输入了不正确的规则而给系统造成安全漏洞。该检测方式不只是适用于本系

统,凡是基于包过滤的防火墙规则存在规则的冲突问题,也同样可以用此方法进行检测。

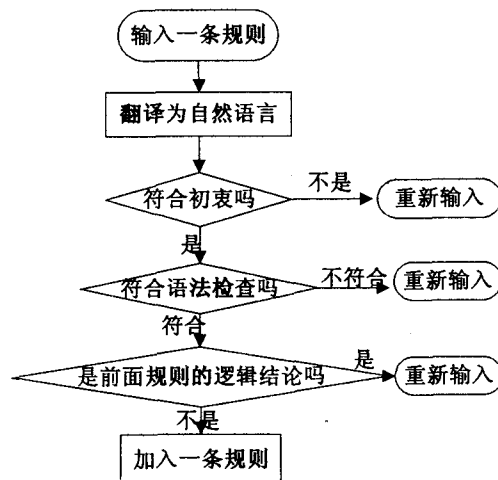


图 3 加入一条规则执行流程

5 结束语

针对 Linux 下的包过滤防火墙,由于利用 Netfilter 中的 iptables 命令配置防火墙规则的复杂性和易错性,在此基础上讨论了防火墙规则的翻译和正确性检测。试验证明它对辅助管理员进行防火墙规则表设置是有一定帮助的。

参考文献:

- [1] 巴卡卡蒂. Red Hat Linux 奥秘[M]. 第 3 版. 北京:电子工业出版社,2000.
- [2] 张 双. 应用代理防火墙中央日志审计子系统的设计与实现[D]. 北京:中国科学院软件研究所,2001.
- [3] Ziegler R L. Linux 防火墙[M]. 北京:人民邮电出版社,2001.
- [4] Callan R. 人工智能[M]. 北京:电子工业出版社,2004.
- [5] 陈 颖. 基于包过滤的个人防火墙的研究与实现[D]. 西安:西安交通大学,2002.
- [5] Paolucci M, Kawamura T, Payne T, et al. Semantic Matching of Web Services Capabilities[C]//First International Semantic Web Conference. Sardinia, Italy; [s. n.], 2002.
- [6] Srinivasan N, Paolucci M, Sycara K. Adding OWL-S to UDDI, implementation and throughput[C]//First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004). San Diego, California, USA; [s. n.], 2004.
- [7] Chao Kuo-Ming, Younas M, Lo Chi-Chun, et al. Fuzzy Matchmaking for Web Services[C]//Proceedings of 19 IEEE Conference on Advanced Network and Information Applications. [s. l.]: IEEE CS, 2005: 721-726.

(上接第 127 页)

参考文献:

- [1] Mooney J K. UDDI technical white paper[EB/OL]. 2002-09-06. http://www.uddi.org/pubs/Uru_UDDI_Technical_WhitePaper.pdf.
- [2] OWL Services Coalition OWL-S: Semantic Makeup for Web Services OWL-S v. 1.1, White Paper[EB/OL]. 2004-11. <http://www.daml.org/services/owl-s/1.1/>.
- [3] OWL-S Semantic Makeup for Web Services[EB/OL]. 2004-11-22. <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/2004/11/22>.
- [4] 胡继才, 万福钧, 吴珍权, 等. 应用模糊数学[M]. 武汉:武汉测绘科技大学出版社, 1998.